https://dx.doi.org/10.22161/ijaers.610.30 ISSN: 2349-6495(P) | 2456-1908(O)

Research on Analytic Algorithm of Building Structure Appearance Based on Improved Learning Grammar

KASSA Anani Thierry[1], Gao Bao Lu[2]

[1]College of Software, Taiyuan University of Technology, China
bigkatiess2@gmail.com
[2]College of Software, Taiyuan University of Technology, China
85389301@qq.com

Abstract**—***Semantic segmentation is one of the biggest and most important concerns of computer vision in order to synthesize novel designs and reconstruct buildings. Traditionally, a human expert was required to write grammars for specific building styles, which limited the scope of method applicability***.** *The main purpose of this paper is to improve learning grammar used for building's façade segmentation. To deal with that, we propose a framework with two layers: in the first layer, we provide a reinforcement learning (RL) techniques to make the segmentation allowing the user to brush strokes on the input image through Gaussian Mixture Models (GMM). Still in this layer, the segmentation can be also make based on shape grammars. Note that for both segmentation, we get as output a ground-truth segmentation. The second layer consist to learn automatically an inferred grammar. Thanks to ground-truth segmentations generated in previous layer, in particular the one generated by RL techniques, we perform clustering techniques to make an improvement of the grammar learned. We evaluate our model on two different datasets and compare in the state-of-the-art our learned-grammar. It show that the proposed outperformed performance gain compared to other learned grammar methods in all the two dataset.*
Keywords**—*Computer vision, Clustering techniques, Gaussian Mixture Models (GMM), learned-grammar, Reinforcement Learning (RL).***

I. INTRODUCTION

How building facades are segmented is great of interest in computer vision due to the number of applications and associated issues such as building information models (BIM). Knowing the regularities in facade layout can be used in video games and movies to generate plausible urban landscapes with realistic rendering [16]. Existing approaches for facade analysis, i.e., the segmentation of facade images into semantic classes, use either conventional segmentation methods or rely on grammar-driven recognition methods [13, 5, 9]. Conventional segmentation methods treat the problem as a pixel labeling task, with the possible addition of local regularity constraints related to building elements, but ignoring the global structural information in the architecture as shown in [26]. On the contrary, methods based on shape grammars impose strong structural consistencies by considering only segments that follow a hierarchical decomposition corresponding to a combination of grammar rules [17, 18].

For a better understanding of our topic, a definition of the term "learning grammars" is essential. There are at least two forms of grammar parsing: the first one is refer to string grammar parsing which consists of an optimal analysis that provides information on the nature of different words and groups of words in the sentence (verbs, nouns, subjects, complements, etc.), it is widely used in Natural Language Processing (NLP) [7]. The second one is called shape grammar parsing that manipulate shapes and their relationships through semantic-geometric rules defined on template shapes (called basic shapes) [7]. It turns out that the groups of words "learning grammar" is nothing more than an automatic learning semantic-geometric rules from images (shapes).

Although Conventional segmentation methods obtain very good pixel-wise scores, these techniques are not appropriate for a number of applications because they frequently produce segments that are inconsistent with basic architectural rules, e.g., irregular window sizes or alignments, or balconies shifted from associated windows. Moreover, as they label only what is visible, ordinary segmentation methods are sensitive to occlusions, e.g., due to potted plants on windows and balconies, or to pervasive

foreground objects in the street: trees, vehicles, pedestrians, street signs, lampposts, etc. As a result, important elements can be partially or totally missing from the produced segments, e.g., portions of wall or even complete windows.

In this work, we focus on structural segmentation, i.e., with global regularities and strict constraints as opposed to just local pixel labeling. More clearly, we propose a new model that combine buildings segmentations and learning grammar. The proposed model consists of two parts: (1) perform a segmentation of a façade building through reinforcement learning techniques and show how shape grammars achieve it too, (2) used clustering algorithm to improve the grammar learned through RL techniques.

This paper is organized as follow: Section 2 gives a brief review of related work. Section 3 details on our approach. The performance of the proposed method is compared with state-of-the-art methods in Section 4. Section 5 summarizes the contributions of this work.

II. RELATED WORK

Combining Computational Geometry with the ideas of Formal Grammars as defined in 1956 by Noam Chomsky in [10], procedural geometry appears first with the definition of L-systems and then with shape grammars. The idea of representing the image contents in a hierarchical and semantized manner can be traced back to the work of Kanade and Ohta [23, 25]. However, the practical applications of grammars to image interpretation or segmentation are attributed to more recent works [4, 21, 24, 11].

In many works, the hierarchical and regular structure of man-made objects is explored to improve segmentation or detection results [21, 24, 11, 19]. In these works, researchers are focused on conventional segmentation techniques.

Conventional segmentation techniques rely on grouping together consistent visual characteristics while imposing piecewise smoothness. Popular methods are based on active contours [15, 6], clustering techniques such as mean-shift [3] and SLIC [1], and graph cuts [2, 7]. Although they obtain very good pixel-wise scores, these techniques are not appropriate for a number of applications because they frequently produce segments that are inconsistent with basic architectural rules. On the contrary, grammar-based methods can infer invisible or hardly visible objects thanks to architecture-level regularity. The use of grammar-based facade parsing has been inspired by the successful application of split grammars for generating virtual urban environments [16]. The key to success is to encode in the grammar basic constraints on the generated objects: the principles of adjacency, non-overlap and snaplines. A number of research works has been aimed at applying the grammar principles for retrieving building models from images [12, 13, 8, 24]. In their work, Teboul et al. present an application of a 2D binary split grammar for parsing rectified facade images [12]. The two kinds of approaches are thus complementary: a better low-level classification or segmentation naturally leads to a better parsing and better overall accuracy (assuming the observed facade follows the architecture style modeled in the grammar).

Although grammatical inference is common in natural language processing (NLP), it is rare in computer vision. Recently, a couple of methods have been proposed to automatically learn shape grammars from ground-truth image annotations [9, 22]. Both operating on split grammars. It seems however this approach does not scale well as the authors have to reduce the size of the training set to keep the induction time practicable. Weissenberg et al. [22] present an alternative technique to learn split grammars from images with ground-truth annotations showing the performance of grammar compression, an experiment in facade image retrieval and examples of virtual façade synthesis.

Previous approaches for shape grammar learning involve a first stage of tree hypothesis generation to produce ground-truth parse trees from the ground-truth segmentation, based on heuristics [9, 22]. In order to get more similar trees in which patterns can be found, Gadde et al. [17, 18] propose to generate these ground-truth parse trees differently, using a small generic handwritten grammar.

III. APPROACH

The proposed model consists of two parts: the first one is to perform a segmentation of façade building through reinforcement learning techniques. This segmentation is formulated in term of Markov Decision process (MDP) using shape grammar convention. Still in this stage, we allow user to brush strokes on the input image for each terminal symbol of the binary split grammar (BSG) with Gaussian Mixture Models (GMM). Through these techniques, we get ground-truth segmentations at this first stage. The output of the first stage become an input for the second stage where we performed hierarchical clustering algorithm to improve the learning grammar. Note that for each architecture image parsed in previous stage it corresponds a ground-truth segmentation thus a binary tree. A set of these binary trees is then parse through the split grammars formalism in 2D. It is then realized a rule compression on these trees by finding and freezing

https://dx.doi.org/10.22161/ijaers.610.30 ISSN: 2349-6495(P) | 2456-1908(O)

repeated subtrees. Furthermore, it is performed clustering on compressed rules to merge inferred rules (learning grammars). These rules are automatically generated by our model and supersedes manual expert work and cuts the time required to build a procedural model of a facade from several days to a few milliseconds. Moreover, thank to inferred rules, it could be designed new buildings, making comparison between two facades architecture, etc. A pipeline of our model is provided in Fig.1.

In the following sections, we will first describe the formalism of shape grammar used in our model (Section 3.1), then will present how we used Markov Decision Process to formulate buildings segmentation through reinforcement learning (Section 3.2) and finally we will describe the clustering techniques used to inferred grammar (Section 3.3).

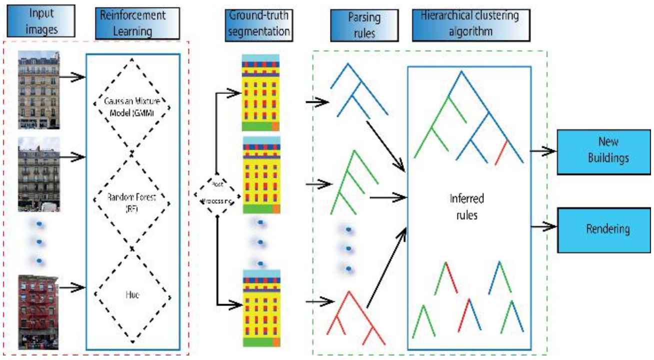

Fig. 1: Overview of our architecture

3.1 Formalism of Shape Grammars

The basic concept of a shape grammar is a labeled rectangle, namely a 5-tuple (c, x, y, w, h), where c is a label or symbol and (c, x, y, w, h) $\epsilon \mathbb{N}^4$ defines the position and dimensions of an axis-aligned rectangle; for notational convenience we may denote a labeled rectangle as c (x, y, w, h). A shape S is a set of labeled rectangles: $S = \{s_1, \cdots, s_n\}$; we will consider these rectangles disjoint.A grammar rule modifies a shape by replacing a labeled rectangle $s_i \epsilon$ S by a set of labeled rectangles (s_i^1,…,s_i^k). In our work we consider only binary split rules (k = 2) that split a labeled rectangle in two along either the horizontal or vertical directions. We denote a rule to break symbol A alongaxis 'h_0' (for horizontal) into symbols B and C as:

$$A(x, y, w, h) \longrightarrow h_{o:\alpha}\{B(x, y, \alpha, h), C(x + \alpha, y, w - \alpha, h)\}$$

The dimensions of B and C are uniquely determined given A, the split direction h_0, and size α, where $\alpha \geq w$; if $\alpha = w$, C is the empty symbol. For brevity we introduce the shorthand notation:

$$A \rightarrow B(\alpha)C$$

which indicates that shape A is split horizontally (↑ means vertically) into a shape of width α and the remainder.

A Binary Split Grammar G is a 4-tuple $(\mathcal{N}, \mathcal{T}, \mathcal{R}, \omega)$, (9) where $\mathcal{N}$ is a set of non-terminals, $\mathcal{T}$ is a set of terminals, ω is a special non-terminal called the axiom and $\mathcal{R}$ a finite set of binary split rules. A labeled rectangle $c(x, y, w, h)$ is terminal if it cannot be further expanded by a rule. To generate a shape S according to a BSG G we start from the axiom $\{\ \omega\ \}$. At each step of the generation a non-terminal element $s_i \epsilon$ S is selected and a rule $r\epsilon$ S applicable to s_i is chosen. After applying r the labeled rectangle s_i is removed from S and replaced by its offspring. This process is called a derivation process and stops when S only contains terminal elements. We call such a shape a segmentation. If the axiom ω corresponds to the image domain, a shape made of terminal elements is an image partition that associates every rectangular region with a label. We can equivalently represent S in terms of a parse (1) tree rooted at ω. During the derivation, the offsprings of s_i are added as its children to the tree. At the end of the process the leaves of the parse tree are terminal elements while its internal nodes represent non-terminal labeled rectangles. The language $L(G)$ is the set of all the possible derivations of the grammar G; in our case this amounts to all possible image segmentations. (2)

3.2 Shape parsing via Reinforcement Learning

In this section, we will introduce in the first time the principles of reinforcement learning and in second time show how we fit these principles to the façade parsing.

➢ ***Principles of Reinforcement Learning*** (5)

In reinforcement learning (RL) [20], an agent interacts with an unknown environment while choosing actions that maximize its cumulative reward. The unknown environment is modeled as a Markov Decision Process (MDP), described by a finite set of states S, a set of actions A, transition probabilities P, and expected rewards R consecutive to actions. At time t, the agent in state s_t, takes action $a_t \epsilon \mathcal{A}(s_t)$ leading the agent to a new state $s_{t+1}$with an immediate reward of r_{t+1}. The tr (6) n state s to s' due to an agent action is subject to the probability $P_{ss'}^a$:

$$P_{ss'}^a = P\ (s_{t+1} = s' | s_t = s, a_t = a)$$

and the reward r_{t+1} received for selecting action a in state s and arriving in state s' is denote by its expectation $R_{ss'}^a$:

$$R_{ss'}^a = E[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']$$

The goal of the reinforcement learning agent is to maximize its long term reward which is:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

The parameter γ is a discount factor and represents how much weight we give to the rewards that we will

https://dx.doi.org/10.22161/ijaers.610.30 ISSN: 2349-6495(P) | 2456-1908(O)

come across in the future. Such a behavior is governed by the agent's policy $\pi(s,a)$, the probability of choosing action a while in state s. This leads to the following state-value function $V^{\pi}(s)$ and action-value function $Q^{\pi}(s,a)$:

$$V^{\pi}(s) = \sum_{a} \pi(s,a)\, Q^{\pi}(s,a)$$

$$Q^{\pi}(s,a) = \sum_{s'} P^{a}_{ss'}(R^{a}_{ss'} + \gamma V^{\pi}(s'))$$

For the most optimal policy π^*, the above two equations lead to the following non-linear Bellman optimality equations:

$$V^{*}(s) = \max_{a} \sum_{s'} P^{a}_{ss'}(R^{a}_{ss'} + \gamma V^{*}(s'))$$

$$Q^{*}(s,a) = \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma \max_{a'} Q^{*}(s',a')]$$

The optimal policy is related to Q^*: to maximize cumulative reward, at every state s, the agent must choose action $a^* = arg\, max_a\, Q^*(s,a)$. An optimal policy is therefore deterministic and derived from Q^*.

- ***Reinforcement Learning for façade parsing***

In order to get a better parsing for façade, our approach is to combine the most techniques used for façade parsing such as: state aggregation, Q-learning and some merits functions. In the following sentences, we describe how each technique is performed and converge to a better parsing.

State aggregation: The first advantage of state aggregation consists in reducing the number of possible states, the second one consists in ensuring consistency along the facade. Instead of such computationally intractable alternatives, we propose to use a common policy over all non-terminals which should be split in a common way. For instance, when splitting floors, the learned policy will depend exclusively on the horizontal coordinate, and not on the height of the floor. This enforces symmetry constraints implicitly, aligning windows across floors, or balconies inside of floors. These advantages come at the price of stochasticity in the decision process. The agent can obtain different rewards, while performing the same action on the same aggregated state. This is why the ability of Reinforcement Learning to cope with stochastic rewards becomes indispensable in our problem setting.

Q-learning: we use a Q-learning agent that iteratively segments facades until converging to an optimal policy. In each episode the agent sequentially builds the segmentation by selecting one rule (action) at a time based on a local information (state). By applying a rule, it may create a terminal symbol, a subtask or a cyclic symbol. Then it receives a reward and reaches a new state where it faces a new decision. The value function is iteratively learned by Q-learning updates. After convergence, reached after around 10^3 episodes, we deterministically parse the facade by following the greedy policy with respect to the estimate of $Q^*(s,a)$. By virtue of being deterministic, and using a policy defined on aggregated states, the delivered parse satisfies symmetry constraints. Moreover, despite the large dimensionality of the original space of states and actions, state aggregation allows us to compactly store the action-value function in a few Mbs of RAM.

Merits functions: The merit functions are defined on the terminals and are involved in the computation of the rewards. If training data is available in the form of segmentation annotations we can obtain supervised merit functions such as ***Random Forest*** (RF) and ***Gaussian Mixture Models*** (GMM) which is based on the RGB values of individual pixels selected by the user through brush strokes on the image for each terminal symbol of the BSG. Both RF and GMM merits are making use of some training examples and therefore require some amount of user interaction. To accommodate also the common case where training data is not available we consider the learning of unsupervised merit functions. In particular for simpler cases where the BSG has only two terminal windows, wall and window, we can separate the two classes based on the heuristic introduced by [14]: the ***hue*** value distinguishes the walls from the windows.

3.3 Clustering to Learning Grammars

This part of our work is linked to previous one, which generated as output the ground-truth labeled images. Based on these outputs, we provide two steps instead of three steps used in previous works [9, 18], leading to generate the learning grammars.

Ground-truth parse trees: a parse tree generation encodes a facade as a binary split tree whose nodes correspond to facade regions, operations and parameters. The parser tries to produce a tree which associate label image matching as much as possible the ground-truth label image. We used generic grammar (Table 1) to generate parse trees. Although it cannot parse real images (in a reasonable time), it is able to successfully parse the ground-truth label images. One advantage of this technique is there are less decisions to make and good choices are tried first [18]. Another advantage is that the generated ground-truth parse trees can be easily understood, as they reuse the same "concepts" and terms as the generic grammar. This translates as well to the specialized grammars that we infer. While generating parse trees using a generic grammar, the number of meta-rules present in the

https://dx.doi.org/10.22161/ijaers.610.30 ISSN: 2349-6495(P) | 2456-1908(O)

trees and thus in ground-truth grammar is bounded by the number of meta-rules in the generic grammar.

Clustering rule patterns: once generated the ground-truth parse tree, the problem we have to deal with here is to define the pattern search as a clustering. The idea is that each given tree or subtree is considered as an object to be grouped with other similar trees or subtrees into clusters. More precisely, given a parse trees $T_1,....,T_n$ covering all the learning set, we want to identify similar subtrees and group them. To deal with that, we use hierarchical clustering algorithm as opposed to LP-based clustering used by [18].

Table 1. Example of generic grammar

Simple generic grammar $\mathcal{G}_{sgen}$		
Axiom	$\xrightarrow{v}$	GroundFloorFloorsRoofFloorsky
GroundFloor	$\xrightarrow{h}$	shop door shop
Floors	$\xrightarrow{v}$	wall (Floorwall)+
Floor	$\xrightarrow{h}$	wall (BalcWinswall)+
Floor	$\xrightarrow{v}$	balconyWinFloor
WinFloor	$\xrightarrow{h}$	wall (windowswall)+
BalcWin	$\xrightarrow{v}$	balconywindow
RoofFloor	$\xrightarrow{v}$	roof (window roof)+

Hierarchical clustering technique is divided into two approaches: bottom-up approach which use first to identify all repeated subtress in individual parse trees separately. The second one is top-down approach used to cluster and merge all parse trees at root level. An example of such a rule merging is shown on Fig.2.

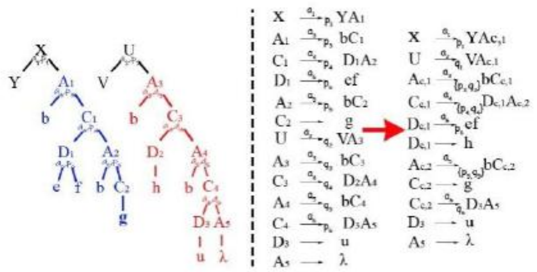

Fig. 2: An example of merging rules.

IV. EVALUATION

In this section, we evaluate our approach based on Reinforcement Learning segmentation in one hand, and in the second hand we evaluate the learning grammar based on hierarchical clustering algorithms. These two approaches are evaluated on two benchmark datasets and compare with state-of-the-art.

4.1 Datasets

We test our model on two benchmarks datasets: ENPC2014 [Raghudeep 2017] with 79 images of Art-deco buildings in Paris and ECP2011 [Teboul2011b] which contain 104 annotated images of Haussmannian buildings in Paris.

4.2 Evaluation based on Reinforcement Learning segmentation

In this section we will show examples of parsing facades using our reinforcement model with specifically rewards as Gaussian Mixture Model (GMM), Random Forest and Hue.

Fig.3: Parsing facades with a 4-color BSG. From left to right: original image, user's brush strokes to train a GMM classifier, pixel-wise segmentation using the GMMs, optimal parse with our algorithm.

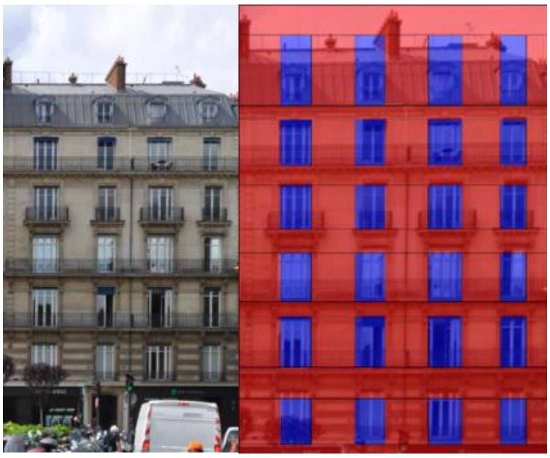

Fig. 4: Parsing facades with Hue reward. On the left the original image, on the right the optimal parse.

https://dx.doi.org/10.22161/ijaers.610.30 *ISSN: 2349-6495(P) | 2456-1908(O)*

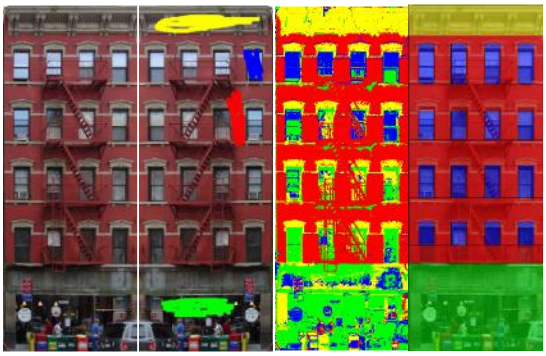

Fig. 5: Parsing facades with Randomized Forest. On the left the original image, on the right the optimal parse.

4.3 Evaluation based on hierarchical segmentation

To do this evaluation, our data are follow some parameters such as: $\mathcal{G}_{\text{gt}}$ (grammar inferred directly from the ground-truth parse trees), $\mathcal{G}_{\text{hcl}}$(grammar inferred directly from hierarchical clustering), in order to show the accuracy of parsing using our learned grammars (Table 2): we report classwise accuracy: average class accuracy, overall pixel accuracy and average intersection-over-union score (IoU). Both datasets ECP2011 and ENPC2014 are segmented and annotated into seven classes: *door, shop, balcony, window, wall, sky and roof*.

Table 2. Segmentation results on the ENPC2014 datasets.

	[Teboul2011b]	[Raghudeep 2017]	$\mathcal{G}_{\text{gt}}$	Ours
Door	49	53	41	**61**
Shop	78	84	78	**89**
Balcony	49	57	46	**65**
Window	51	59	46	**68**
Wall	72	79	78	**88**
Sky	97	96	95	**95**
Roof	52	54	49	**62**
Average	64.1	68.9	61.8	**74.5**
Overall	68.4	74.3	69.5	**79.8**
IoU	48.0	57.8	48.2	**60.4**

Furthermore we show few visual segmentations using our learned grammar with number of episodes for convergence and segmentation accuracy.

(624, 90.1%) (500, 92.2%)

Fig. 6: Qualitative results on ECP2011 dataset. Image (left) and segmentation using learned grammar $\mathcal{G}_{hcl}$ (right) are shown here along with number of episodes for convergence and segmentation accuracy.

(804, 80.4%) (648, 85.1%)

Fig. 7: Qualitative results on ENPC2014 dataset. Image (left) and segmentation using learned grammar $\mathcal{G}_{hcl}$ (right) are shown here along with number of episodes for convergence and segmentation accuracy.

V. CONCLUSION

In this paper, we improve the learning grammar through a hierarchical clustering algorithm. We demonstrated that hierarchical clustering technique outperform façade segmentation through bottom-up approach which use first to identify all repeated subtress in individual parse trees separately and the top-down approach used to cluster and merge all parse trees at root level. We achieved state-of-the-art performance on a challenging benchmark, and showed the potential of the method to deal with a wide variety of buildings.

VI. ACKNOWLEDGMENTS

We thank Bao Gao Lu for helpful discussion, advises in high-performance computing.

VII. REFERENCES

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(11), 2274{2282 (2012)

https://dx.doi.org/10.22161/ijaers.610.30 ISSN: 2349-6495(P) | 2456-1908(O)

[2] Berg, A.C., Grabler, F., Malik, J.: Parsing images of architectural scenes. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1{8. IEEE (2007)
[3] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(5), 603{619 (2002)
[4] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing with attribute graph grammar. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(1):59–73, 2009.
[5] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof. Irregular lattices for complex shape grammar facade parsing. In CVPR, 2012.
[6] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International journal of computer vision 1(4), 321{331 (1988)
[7] Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(2), 147{159 (2004)
[8] M. Mathias, A. Martinovic, J. Weissenberg, and L. V. Gool. Procedural 3D building reconstruction using shape grammars and detectors. In 3DIMPVT, 2011.
[9] Martinovic, A., Van Gool, L.: Bayesian grammar learning for inverse procedural modeling. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 201{208. IEEE (2013)
[10] Noam Chomsky. Three Models for the description of Language. In IRE Transactions on information theory, 1956.
[11] N. Ahuja and S. Todorovic. Connected Segmentation Tree - A Joint Representation of Region Layout and Hierarchy. In CVPR, 2008.
[12] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In CVPR, pages 3105–3112, 2010.
[13] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. In CVPR, pages 2273–2280, 2011.
[14] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakisand N. Paragios. Parsing facades with shape grammars and reinforcementlearning. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 7, pages 1744–1756, 2013.
[15] Osher, S., Paragios, N.: Geometric level set methods in imaging, vision, and graphics. Springer (2003)
[16] P. Muller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. ACM Transations on Graphics, 25(3):614–623, 2006.
[17] Raghudeep Gadde, Renaud Marlet and Nikos Paragios. Learning Grammars for Architecture-Specific Facade Parsing. International Journal of Computer Vision, pages 1–27, 2016.
[18] Raghu Deep Gadde. Semantic Segmentation of Highly Structured and Weakly Structured Images. Signal and Image Processing. Université Paris-Est, 2017.
[19] Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1640{1647. IEEE (2012)
[20] Sutton R.S. and A.G. Barto. Introduction to reinforcement learning. MIT Press, 1998.
[21] W. Wang, I. Pollak, T.-S. Wong, C. A. Bouman, and M. P. Harper. Hierarchical stochastic image grammars for classification and segmentation. IEEE Transactions on Image Processing, 15:3033–3052, 2006.
[22] Weissenberg, J., Riemenschneider, H., Prasad, M., Van Gool, L.: Is there a procedural logic to architecture? In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 185{192. IEEE (2013)
[23] Y. Ohta, T. Kanade, and T. Sakai. A production system for region analysis. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pages 684 – 686, 1979.
[24] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In CVPR (2), pages 2145–2152, 2006.
[25] Y. Ohta, T. Kanade, and T. Sakai. An analysis system for scenes containing objects with substructures. In Proceedings of the Fourth International Joint Conference on Pattern Recognitions, pages 752–754, 1978.
[26] David Ok, Mateusz Koziński, Renaud Marlet, Nikos Paragios. High-Level Bottom-Up Cues for Top-Down Parsing of Facade Images. 3DIMPVT, Oct 2012, Zürich, Switzerland. pp.N/A, 2012.